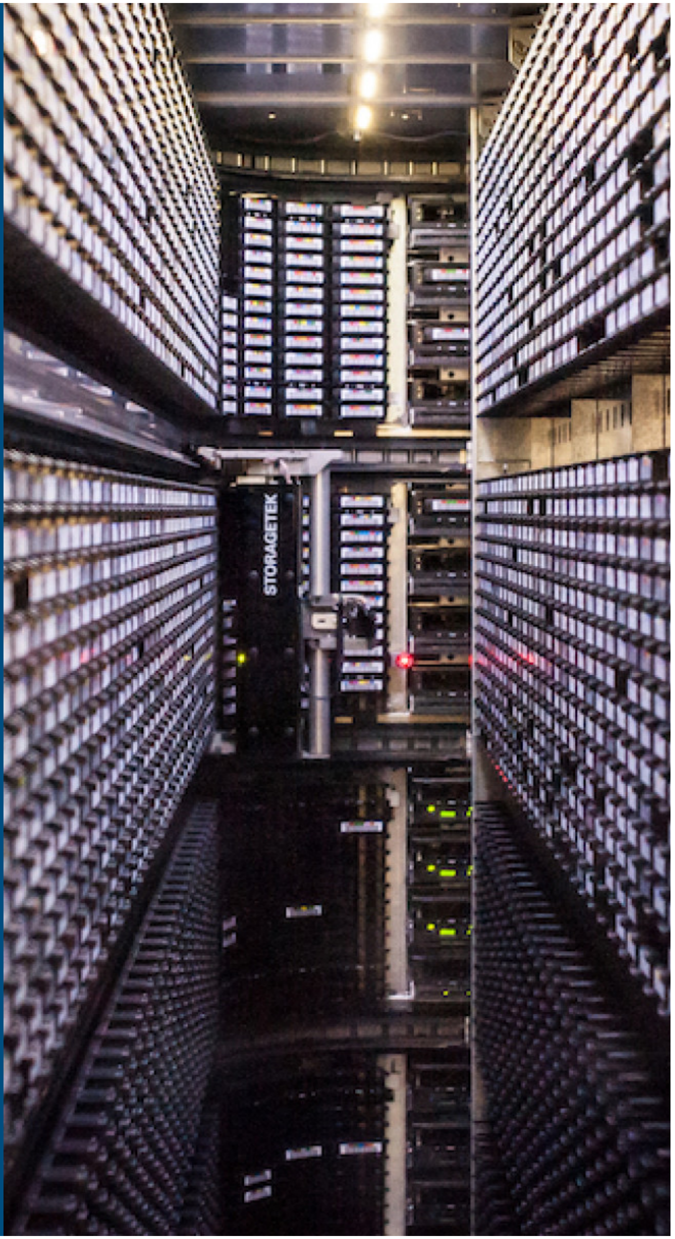
ASTRON

Netherlands Institute for Radio Astronomy

LOFAR use cases for DAC21

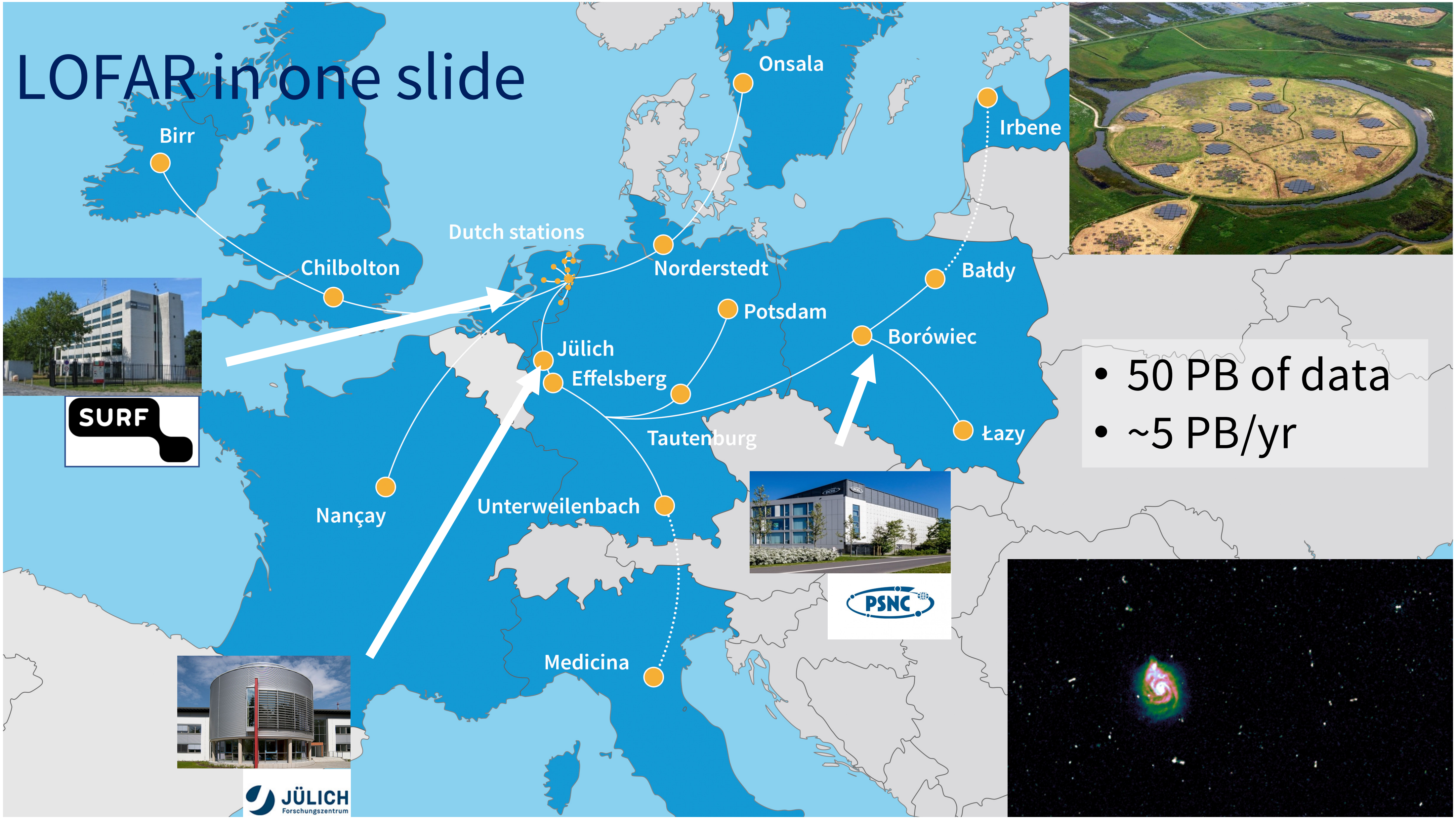
Yan Grange, Vishambhar Nath Pandey

Netherlands Institute for Radio Astronomy



LOFAR in one slide

- 50 PB of data
- ~5 PB/yr



Use cases

Label:name	Steps
UC1: ingest	• Copy data to ingest RSE • Register with Rucio • Define QoS-based life cycle rules • Download file(s) using Rucio
UC2a: imaging	• Define structure (data sets and containers) • Download “processing package” • Create image • Put image in the data lake, with some life cycle info
UC2b: imaging (continued)	• Query the Virtual Observatory for the souce imaged in 2a • Put in ESAP shopping basket • Query data lake for image from 2a and put in shopping basket • Use DLaaS notebook to make a composite image

Of course, you can read all about this in the Notebook ☺

LOFAR

ASTRON
Netherlands Institute for Radio Astronomy

UC1 - result

- Used rclone to copy data from local disk to a non-deterministic end point
  - Started with PIC-inject (thanks for making it available to us!), but bandwidth wasn’t too great from our SARA node
  - Kudos to Paul for setting up one at DESY for us.
- Registering data to Rucio, and applying rules through python bindings:
- Ran into some issues with OIDC tokens
  - Large files -> While the script runs the OIDC token times out
- Wrote a function running in the background that rus oidc-token once in a while (minute) and write it to where Rucio expects it.
  - Sometime token did not work in rclone for some odd reason. In the end this seemed to be a known issue

https://git.astron.nl/astron-sdc/escape-wp2/dac21-ingest

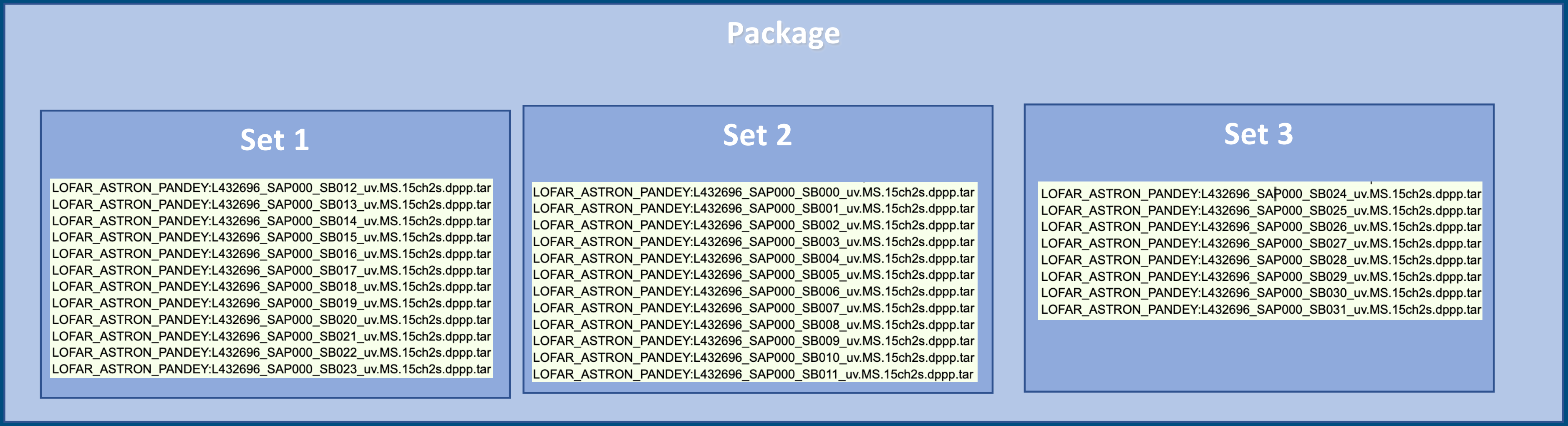
LOFAR

ASTRON

Netherlands Institute for Radio Astronomy

UC2a - result

- Typically a radio image is made by calibrating a calibrator and applying the solutions to the target. So the goal here is to split an observation in three parts, and make them accessible as one object.



Package

Set 1

LOFAR_ASTRON_PANDEY:L432696_SAP000_SB012_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB013_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB014_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB015_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB016_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB017_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB018_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB019_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB020_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB021_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB022_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB023_uv.MS.15ch2s.dppp.tar

Set 2

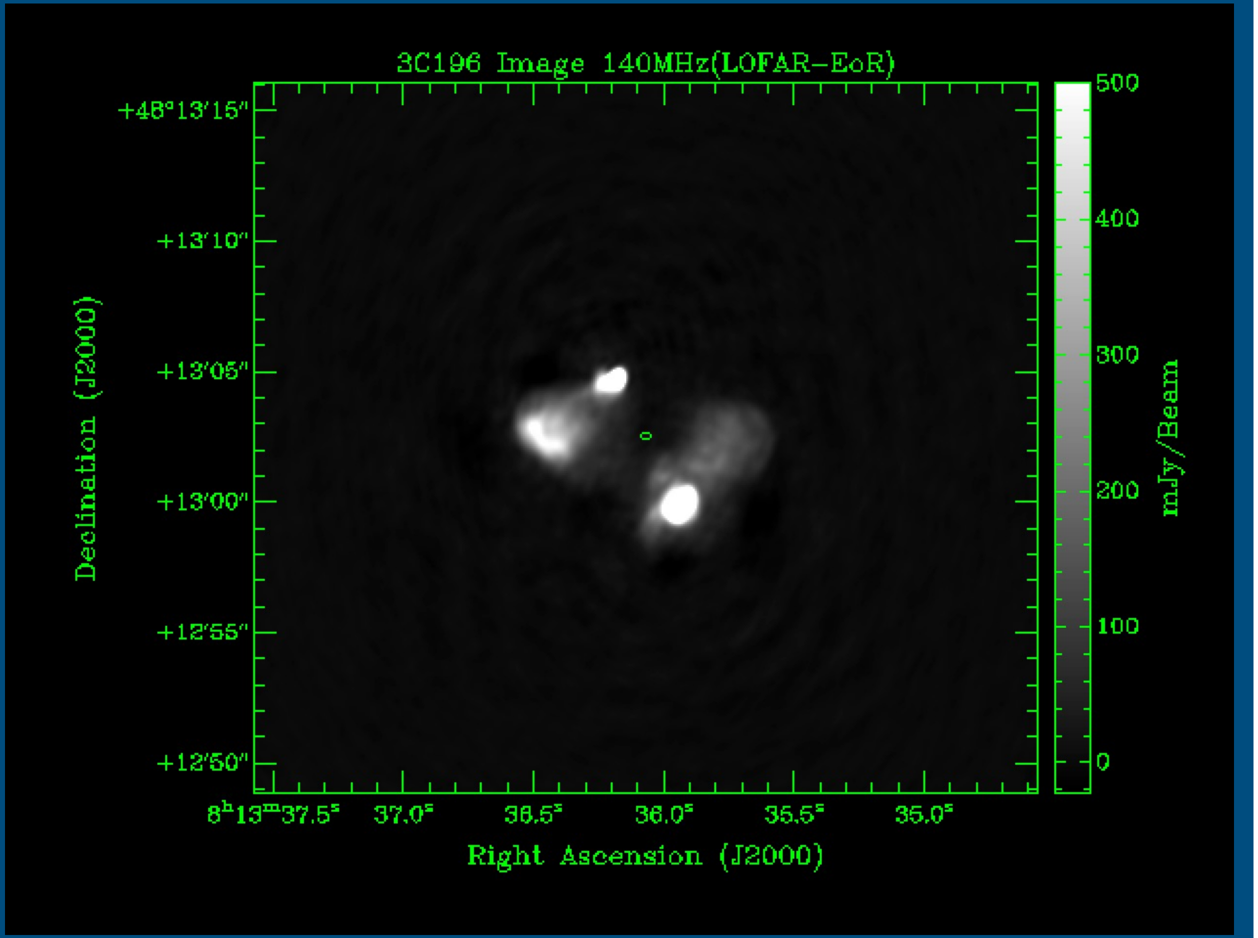
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB000_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB001_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB002_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB003_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB004_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB005_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB006_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB007_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB008_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB009_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB010_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB011_uv.MS.15ch2s.dppp.tar

Set 3

LOFAR_ASTRON_PANDEY:L432696_SAP000_SB024_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB025_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB026_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB027_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB028_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB029_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB030_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB031_uv.MS.15ch2s.dppp.tar

UC2a - result

- We got a LOFAR image! Just bragging a bit: that’s a sub arc-second resolution radio image!
- Uploaded FITS version to the Datalake
- Pretty much everything went according to plan



LOFAR

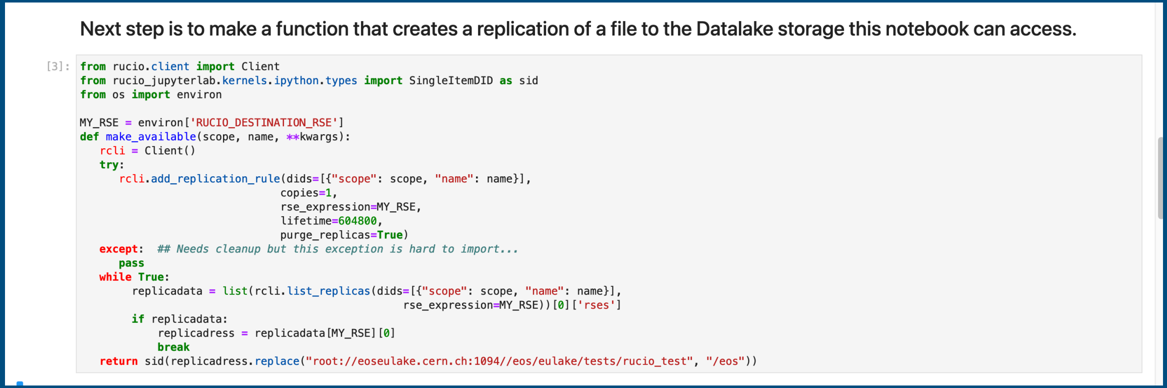
ASTRON
Netherlands Institute for Radio Astronomy

UC2b - result

- Found Hubble observations of 3C196 in the Virtual Observatory (this is a pretty dim source in optical as far as I can judge).
  - Sent data to ESAP using the SAMP protocol, and Rucio data sets in same shopping basket.
- Fire up DLaaS notebook
  - Install ESAP shopping basket client
  - Pull in shopping basket data from VO
  - Move data to FUSE mounted folder using rule
- Create combined image

UC2b – result (2)

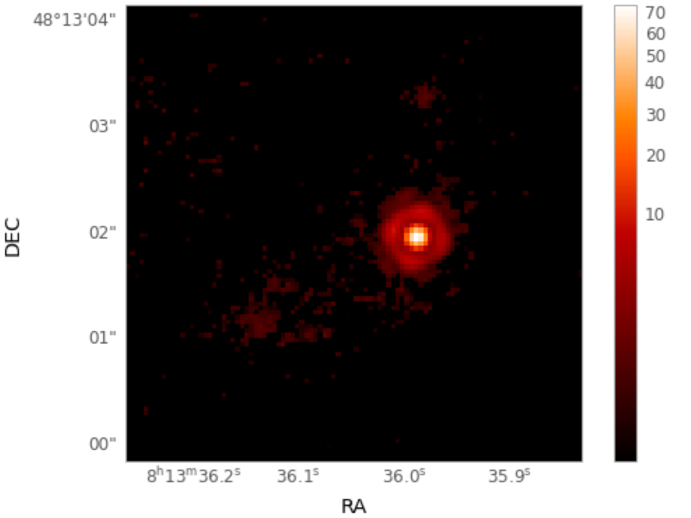
- Top: Hubble image, bottom: combined image (LOFAR image with Hubble contours_)
- Integration between ESAP and DLaaS may need some extra discussion on what to expect where.
- Code to make available is not too complex:

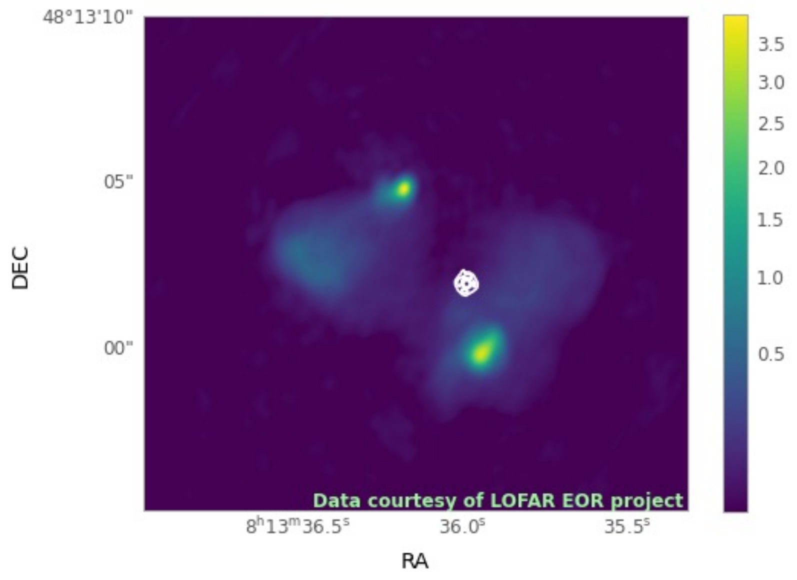


Next step is to make a function that creates a replication of a file to the Datalake storage this notebook can access.

```
[3]: from rucio.client import Client
     from rucio_jupyterlab.kernels.ipython.types import SingleItemDID as sid
     from os import environ

     MY_RSE = environ['RUCIO_DESTINATION_RSE']
     def make_available(scope, name, **kwargs):
        rcli = Client()
        try:
           rcli.add_replication_rule(dids=[{"scope": scope, "name": name}],
                                    copies=1,
                                    rse_expression=MY_RSE,
                                    lifetime=604800,
                                    purge_replicas=True)
        except:  ## Needs cleanup but this exception is hard to import...
           pass
        while True:
             replicadata = list(rcli.list_replicas(dids=[{"scope": scope, "name": name}],
                                                   rse_expression=MY_RSE))[0]['rses']
             if replicadata:
                 replicadress = replicadata[MY_RSE][0]
                 break
        return sid(replicadress.replace("root://eoseulake.cern.ch:1094//eos/eulake/tests/rucio_test", "/eos"))
```





Data courtesy of LOFAR EOR project

LOFAR

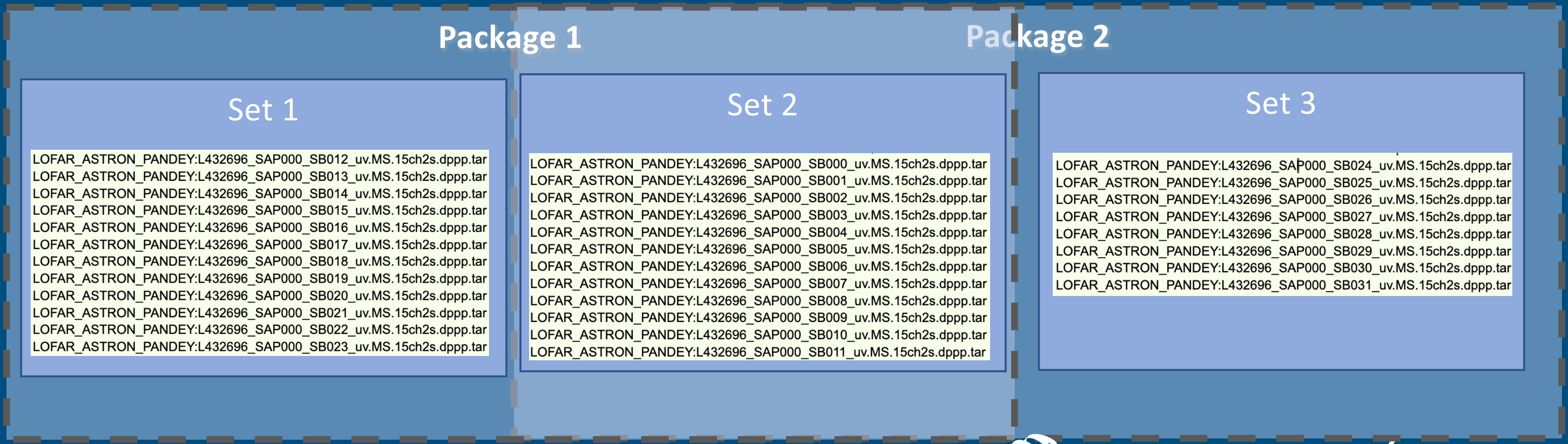
ASTRON
Netherlands Institute for Radio Astronomy

Use case conclusions

Label:name	Steps	Conclusion
UC1: ingest	• Copy data to ingest RSE external to Rucio • Register with Rucio • Define QoS-based life cycle rules • Download file(s) using Rucio	• This did not work too well but is external to the Data Lake. • Python bindings work well for this! • Rules work well. Issues may arise when deleting files in webdav that have rules on them (not too surprising of course) • That worked as expected ☺
UC2a: imaging	• Define structure (data sets and containers) • Download “processing package” • Create image • Put image in the data lake, with some life cycle info	• All steps were executed as we expected!
UC2b: imaging (continued)	• Query the Virtual Observatory for the souce imaged in 2 and put in ESAP shopping basket • Query data lake for image from 2a and put in shopping basket • Use DLaaS notebook to make a composite image	• ESAP query was not possible -> used VO tools and the SAMP protocol to put in shopping basket • ESAP query worked as expected • Success, but with some extra code

Extra grouping experiment

- After DAC21 we also performed the experiment with a dataset that is part of multiple ‘observing packages’.
  - This totally works as expected ☺



Set 1

LOFAR_ASTRON_PANDEY:L432696_SAP000_SB012_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB013_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB014_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB015_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB016_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB017_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB018_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB019_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB020_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB021_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB022_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB023_uv.MS.15ch2s.dppp.tar

Set 2

LOFAR_ASTRON_PANDEY:L432696_SAP000_SB000_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB001_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB002_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB003_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB004_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB005_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB006_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB007_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB008_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB009_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB010_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB011_uv.MS.15ch2s.dppp.tar

Set 3

LOFAR_ASTRON_PANDEY:L432696_SAP000_SB024_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB025_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB026_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB027_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB028_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB029_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB030_uv.MS.15ch2s.dppp.tar
LOFAR_ASTRON_PANDEY:L432696_SAP000_SB031_uv.MS.15ch2s.dppp.tar

LOFAR

ASTRON
Netherlands Institute for Radio Astronomy

Lessons learned

- register-after-upload flag is quite useful (essential?) when uploads have a high chance of failing.
- Deleting the last copy of data can be a challenge if the data is actually corrupted
  - Normally one adds a rule to protect the data for a second, but if the data does not exist, the rull will immediately get stuck.
- Very hard to get a grip on data that exists “behind Rucios back” as in UC1.

Open questions

- How much to share, how much to manage ourselves
  - FTS? Rucio?
- Directory structure in downlads:
  - Basically now this is SCOPE/toplevelcontainer/file now, but we may want all the containers in between. Is that possible?
- Fully public data
  - Without any AAI, VO’s etc.
  - How does that intergate in EOSC at all?
- Data crossing the VO boundary?
- Can we show that DLaaS scales to other storage systems, places?

LOFAR ASTRON Netherlands Institute for Radio Astronomy

Forward (in WP2)

- LOFAR LTA as an external RSE
  - Public data for users who are a member of any VO and want LOFAR data
  - As an example of dding an external ‘legacy’ archive to a data lake
- Batch processing in ESAP: looking at it with lots of interest.
- DLaaS integration in ESAP: access to the shopping basket
  - I threatened/promised to start up some discussion about that. Haven’t pushed it through but consider it a main activity of the integration with the compute ☺
- Would really like to see DLaaS deployed to (at least) one other site (from the LOFAR perspective I woukd say: maybe dCache?).
- We’ll (finally) set up our own instance to play with (it is planned now!)

LOFAR ASTRON Netherlands Institute for Radio Astronomy

Forward look

- Embargoed data -> that is a very relevant discussion to us too!
- Integration with the IVOA standards
  - Making Rucio data data available from VO tooling and processing through ESAP
- Had a chat with Ron on CS3MESH4EOSC on their work. Really looking forward to what comes out of that!
- Potentially interesting future topics:
  - Multihop/complex network transfers (e.g. when a processing system is behind a firewall or for using optimal network connections).
  - Messaging system, e.g. when QoS transition (tape->disk) is done, warn the processing pipeline to initiate the download. Or maybe have multistep rules?

LOFAR ASTRON Netherlands Institute for Radio Astronomy